Agent-Based CBR for Decision Support System

Mythili Devi Nagaiah

Research Scholar, Department of Computer Science and Applications
SRM University, Kattankulathur, Chennai, India

Abstract — The aim of this paper is to describe about Case Based Reasoning (CBR) which is based on agents and the implementation in Decision Support System (DSS). The Introduction Section gives a introduction about Data Mining, integration of Data Mining concepts with CBR, and defines the characteristics and process cycle of CBR. The Second Section describes about the Agents, DSS, and Agent-Based DSS. The Third Section describes about the CBR in Decision Support System. The Fourth Section describes the CBR (Agent-Based) for Decision Support System and Interaction between CBR Agents and components of Decision Support System. The Final section gives a conclusion about the paper.

Index Terms— CBR, Data Mining, Integration of Data Mining and CBR, Agents, Decision Support System,

1. INTRODUCTION

A. Data Mining

Data Mining (DM) has become a popular method for extracting information from large databases. During the last few years, new technology has reduced the cost of storing data, and better technology in database management has made it easier to handle databases with gigabytes or terabytes of data. Most enterprises, organizations and governments now have huge databases, and the focus is thus changing from data collection to data analysis. If the size of the data base, dimension and the complexity of data representation are given, even then it is difficult for human to analyse the data.

B. Case Based Reasoning

Case-Based Reasoning (CBR) is a method for solving problems by comparing a problem situation to previously experienced ones. The aim is to store information about earlier situations, and when new ones arrive, find the situation that is most similar, and reuse it to match the new problem if the most similar problem does not match sufficiently. This may involve using background knowledge or asking a user.

Information about the problem solving experience is learned by the Case-Based Reasoning system and the aim is to be able to handle an increased number of situations and also reason more on each situation to certify that it is handled correctly. Each case is a set of features, or attribute-value pairs, that encode the context in which the ambiguity was encountered. The case retrieval algorithm is mostly a simple k-nearest neighbour's algorithm. The basic case-based learning algorithm performs poorly when cases contain many irrelevant attributes. Unfortunately, deciding which features are important for a particular learning task is difficult.

At the highest level of generality, a general CBR cycle may be described by the following four processes: [1]

CBR methods can be divided into four steps, retrieve - find the best matching of a previous case, reuse - find what can be reused from old cases, revise - check if the proposed solution is correct, and retain - learn from the problem solving experience.

1) Retrieve

Retrieving a case means to start with a (partial) new case, and retrieve the best matching previous case. It involves the following subtasks:

Identify features - this may simply be to notice the feature values for a case. This can be filtering out noisy problem descriptors, infer other relevant problem features, check whether the feature values make sense in the given context, or generate expectations of other feature values.

Initially match - usually done in two parts, first an initial matching process which gives a list of possible candidates, which are then further examined to select the best. There are three ways of retrieving a case or a set of cases: By following direct index pointers from the problem features, by searching an index structure, or by searching in a model of domain knowledge.

Select - select a best match from the cases returned by the initially match. The reasoner tries to explain away non-identical features. If the match is not good enough, a better one is sought by using links to closely related cases. The selection process can generate consequences and expectations from each retrieved case, by using an internal model or by asking the user.

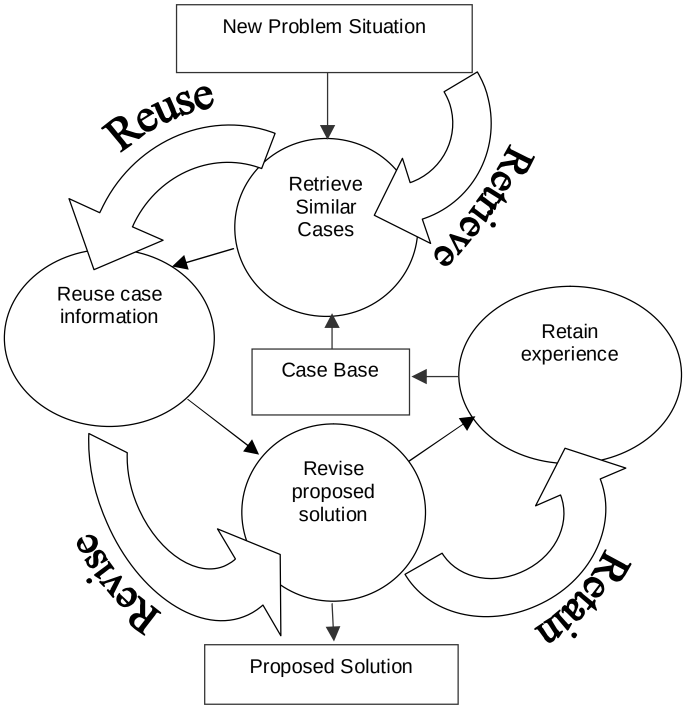

Figure 1 The Case Based Reasoning Processing Cycle

2) Reuse

The focus of reuse is to find the difference between the new and the old case, and find what part of the old case that can be used in the new case. It either involves copying the old solution or adapting it:

Copy - in simple classification, the differences between the old and new case are abstracted away, and the solution is simply copied from the old case.

Adapt - either the solution itself can be (transformed and) reused, or the past method that produced the solution can be used.

3) Revise

If the solution generated by the last phase is not correct, the system can learn from its failures. This involves:

Evaluate - try the solution proposed by the reuse-phase in the real environment, and evaluate it.

Repair fault - If the solution evaluated badly, find the errors or flaws of the solution, and generate explanations for them.

4) Retain

Incorporate what is useful to learn from the problem solving experience into the existing knowledge.
Sub processes are:

Extract - if the problem was solved using an old case, the system can build a new case, or generalize an old case to include the new case as well. If the user was asked, a new case should be constructed. Explanations may be included in the case.

Index - decide what types of indexes to use for future retrieval.

Integrate - modify the indexing of existing cases after the experience, strengthen the weight of features that were relevant, and decrease the weight of features that lead to retrieval of irrelevant cases.

C. Data Mining and Case Based Reasoning Integration

Both methods are used for decision support, to organize and process information to make it available for improving the quality of decisions. The decisions might be taken by humans within an organization or by a computer system.

CBR relies heavily on the quality of the data collected, the amount of data, the amount of background knowledge and a way of comparing cases to decide which is most similar. The method is best suited for domains that change, and where we have little knowledge of underlying processes that govern the domain.

Data Mining is a way of extracting information from databases and can thus be used for extracting information which is relevant for a problem situation. It could also be used to find ``unexperienced'' problem situations from a database and represent it as a case, possibly by interacting with a user. Data Mining can infer rules, classifications and graphs from the data which can be used as background knowledge in a CBR system, and also to compute the similarity between cases.

Some Data Mining algorithms require background knowledge, which can be taken from a CBR system. By integrating the methods, we hope to make better use of information, and that this can lead to a growth for computer systems as well.

2. AGENTS AND AGENT BASED DSS

2.1. Agents

An Agent is an idea, or concept provides a convenient way to describe software entity. Various authors have proposed different definitions of agents; these commonly include concepts such as

- *Persistence* (code is not executed on demand but runs continuously and decides for itself when it should perform some activity)
- *Autonomy* (agents have capabilities of task selection, prioritization, goal-directed behavior, decision-making without human intervention)
- *Social ability* (agents are able to engage other components through some sort of communication and coordination, they may collaborate on a task)
- *Reactivity* (agents perceive the context in which they operate and react to it appropriately).

The *Agent* concept is most useful as a tool to analyze systems. The concepts mentioned above often relate well to the way we naturally think about complex tasks and thus agents can be useful to model such tasks

2.2. Decision Support System

Decision support system is an interactive computer-based system intended to help decision makers use communications technologies, data, documents, knowledge and/or models to identify and solve problems, complete decision process tasks, and make decisions. The concept of decision automation is deceptively simple and intriguingly complex. From a narrow perspective, a decision is a choice among defined alternative courses of action. From a broader perspective, a decision involves the complete process of gathering and evaluating information about a situation, identifying a need for a decision, identifying or in other ways defining relevant alternative courses of action, choosing the 'best', the 'most appropriate' or the 'optimum' action, and then applying the solution and choice in the situation.

2.3. Agent Based DSS

In knowledge engineering, agents offer the flexibility to integrate many different categories of processing within a single system. Agent definitions range from descriptions based on a functional analysis of how agents are used in technology to far more ranging expositions based on different interpretations of the role and objectives of artificial intelligence and cognitive science. Artificial intelligence is a very diverse field and agents are used as metaphors for work in many areas.

Multi-agent systems are appropriate for domains that are naturally distributed and require automated reasoning. Agents should perform the following capabilities to some degree:

- Planning or reacting to achieve goals,

- Modelling the environment to properly react to situations,

- Sensing and acting,

- Inter-agent coordination,

- Conflict resolution (coordination is a continuous process; conflict resolution is event-driven, triggered by conflict detection).

To design a multi-agent system for a given problem, the designer has to understand how should agent and AI techniques be applied to the domain, what competencies agents need, and which techniques implement those competencies. Thus, multi-agent system design consists of

(1) Dividing resources and domain responsibilities among agents,

(2) Determining which core competencies satisfy which domain responsibilities, and

(3) Selecting techniques to satisfy each core competency.

According to distributed domain-specific responsibilities agent-based systems may be heterogeneous, with each agent responsible for a different set of goals or homogeneous, where agents share the same goals. Agents in the proposed system work according to simple workflow that is specified by user in terms of required support. [6]

Decision support systems are used by people who are skilled in their jobs and who need to be supported rather than replaced by a computer system. The broadest definition states that decision support system is an interactive computer-based system or subsystem intended to help decision makers use communications technologies, data, documents, knowledge and/or models to identify and solve problems, complete decision process tasks, and make decisions. Five specific decision support system types include:

- Communications-driven DSS,

- Data-driven DSS,

- Document-driven DSS,

- Knowledge-driven DSS,

- Model-driven DSS.

3. CBR IN DECISION SUPPORT SYSTEM

CBR is concerned with the solution of problems by identifying and adapting similar problems stored in a library of past experience/solutions. In our approach, CBR techniques are encapsulated into a Stimulus Agent, which dynamically influences the decision making process by assuming the dual roles of a *story teller* and an *advise.* [2]

In the *story teller* role, the CBR agent helps in gaining alternative perspectives on the problem at hand by intervening autonomously during a decision making session and exposing them interactively to how problems are different, but relevant contexts have been approached and solved. The main two functions of the story teller role are to stimulate lateral thinking (associative thinking affecting the way problems are perceived and structured), to serve as a tutoring aid to inexperienced.

In the *advisory* role, the CBR agent attempts to exploit the knowledge stored in the case library to provide specific, context-related suggestions during the decision making process. In both cases, the CBR agent provides suggestive guidance to replace the judgment.

While many CBR systems have been designed for case-based problem solving, relatively fewer systems have been designed for aiding decision making processes in which human judgment plays a central role. These systems tend to embody an outcome oriented view in which the emphasis is on *why* a solution works for a particular problem, ignoring the process by which the solution is obtained. [3]

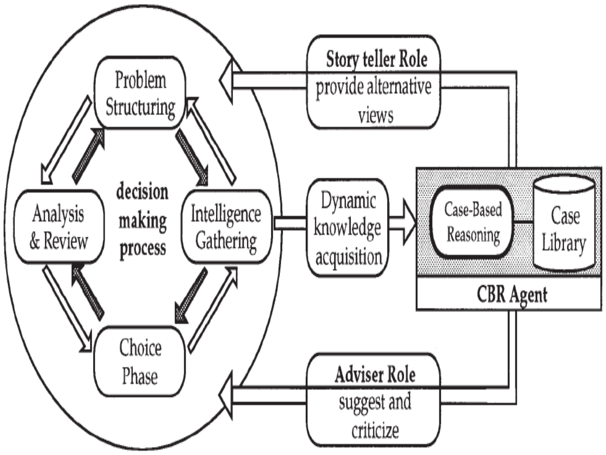

Figure 2 Impact of CBR Agent in decision making process

4. INTERACTION BETWEEN CBR AGENTS AND COMPONENTS OF DSS

The kernel is the main part of the DSS and has as role to manage all the decision-support process. The environment includes essentially the actors and Distributed Information Systems (DIS) and feeds permanently the system with information describing the state of the current situation. In order to apprehend and to deal with this information, specific knowledge related to the domain as ontology and proximity measures are required. The final goal of the DSS is to provide an evaluation of the situation by comparing it with past experimented situations stored as scenarios in a Scenario Base (SB). The kernel is a MAS operating on three levels. It intends to detect significant organizations that give a meaning to data in order to support finally the decision making. [4]

4.1. Situation representation:

One fundamental step of the system is to represent the current situation and its evolution over time. Indeed, the system perceives the facts that occur in the environment and creates its own representation of the situation thanks to a factual agent's organisation. This approach has as purpose to let emerge subsets of agents.

4.2. Situation assessment:

This mechanism is studied "manually" by an expert of the domain and is similar to a Case-Based Reasoning (CBR), except it is dynamic and incremental. According to the application, one or more most pertinent scenarios are selected to inform decision-makers about the state of the current situation and its probable evolution, or even to generate a warning in case of detecting a risk of crisis. The evaluation of the situation will be then reinjected in the perception level in order to confirm the position of the system about the current situation. This characteristic is inspired from the feedbacks of the natural systems. In that manner, the system learns from its successes or from its failures.

4.3. Automating decisions:

Outcomes generated by the assessment agents are captured by a set of performative agents and are transformed in decisions that may be used directly by the final users.

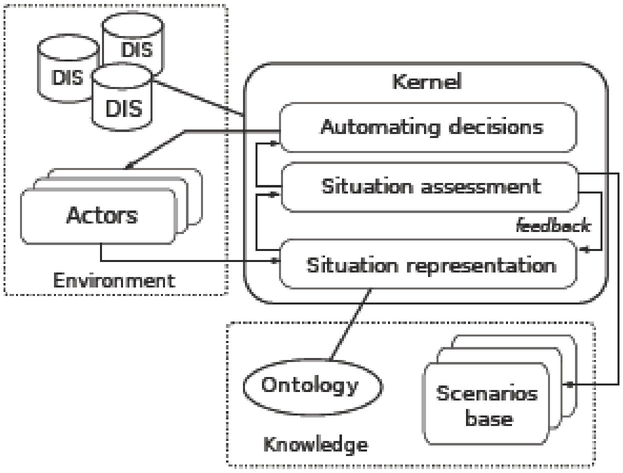

Figure 3 Architecture of Decision Support System

5. CONCLUSION

We have described in this paper an agent-based approach which is also based on Case Reasoning that aims to build a DSS. The system intends to help emergency planners to detect risks and to manage crisis situations by perceiving, representing and assessing a current situation. Decision Support Systems (DSS) are the need of the hour to assure results at a faster rate that best match the buyers' preferences and give valid recommendations. The Case-Based Reasoning (CBR) approach is used, which is a novel paradigm that solves a new problem by remembering a previous similar situation and reusing the information on and knowledge of that situation to bring out similar cases at a faster rate. With the rapid development of case-based reasoning (CBR) techniques, CBR has been widely applied to real-world applications such as agent-based systems for ship collision avoidance. A successful Agent Based CBR relies on a high-quality case base.

REFERENCES

[1] Althoff, K (2001), "Case-Based Reasoning", ICCBR-01 Workshop On Case-Based Reasoning Authoring Support Tools, 31st July.

[2] Albert A. Angehrn and Soumitra Dutta, "Case – Base Decision Support"

[3] Christian Bohm, Maria Rosa, Omor Choitti,(2003), " Data Mining Opportunities and Challenges", P.No. 421-426

[4] Fahem Kebair and Frederic Serin, " Agent Oriented Approach for Detecting and Managing Risks in emergency Situations"

[5] Hao Lang Zhang, "Agent Based Open Connectivity for Decision Support System"

[6] Jolana Sebestyenova, "Case-based Reasoning in Agent-based Decision Support System"

[7] Kolodner, J. Case-Based Reasoning. Morgan Kaufman, 1993.

[8] Manheim, M.L. An architecture for active DSS. Proceedings of the 21st Annual Hawaii International Conference on System Sciences. IEEE Computer Society Press

[9] Wettschereck D., D. W. Aha: Weighting Features. Proc. of the First Int. Conference on Case-based Reasoning, 1995